ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

MÔN HỌC: VI XỬ LÝ - VI ĐIỀU KHIỂN (TN) (CO3010)

LAB 4 - SCHEDULER

Giảng viên hướng dẫn: Lê Trọng Nhân
Tôn Huỳnh Long

Lớp: L02

Sinh viên thực hiện: Võ Thị Ánh Tuyết - 2213826

Thành phố Hồ Chí Minh, Tháng 09 Năm 2025

Contents

1 Exercise 1

In this exercise, I will add one more file named global.c and apply scheduler for Lab3

```c
#include "global.h"

sTask SCH_tasks_G[SCH_MAX_TASKS];
int status = AUTO_INIT;

unsigned char SCH_Add_Task(void (*pFunction)(), unsigned int DELAY
    , unsigned int PERIOD)
{
  uint32_t Index = 0;
  // First find a gap in the array (if there is one)
  while ((SCH_tasks_G[Index].pTask != 0) && (Index < SCH_MAX_TASKS
   ))
  {
    Index++;
  }
  if (Index == SCH_MAX_TASKS) {
    return 1; // Also return an error code
  }
  // If we're here, there is a space in the task array
  SCH_tasks_G[Index].pTask = pFunction;
  SCH_tasks_G[Index].Delay = DELAY;
  SCH_tasks_G[Index].Period = PERIOD;
  SCH_tasks_G[Index].RunMe = 0;
  return Index; // return position of task (to allow later
   deletion)
}

void SCH_Update(uint32_t Index)
{
  if (SCH_tasks_G[Index].pTask) {
    if (SCH_tasks_G[Index].Delay == 0) {
      // The task is due to run
      SCH_tasks_G[Index].RunMe += 1; // Inc. the 'RunMe' flag
      if (SCH_tasks_G[Index].Period) {
        // Schedule periodic tasks to run again
```

```
33          SCH_tasks_G[Index].Delay = SCH_tasks_G[Index].Period;
34        }
35      }
36      else
37      {
38        // Not yet ready to run: just decrement the delay
39        SCH_tasks_G[Index].Delay -= 1;
40      }
41    }
42  }
43
44  unsigned char SCH_Delete_Task(uint32_t TASK_INDEX)
45  {
46    int Return_code;
47    if (SCH_tasks_G[TASK_INDEX].pTask == 0) Return_code = 1;
48    else Return_code = 0;
49    SCH_tasks_G[TASK_INDEX].pTask = 0x0000;
50    SCH_tasks_G[TASK_INDEX].Delay = 0;
51    SCH_tasks_G[TASK_INDEX].Period = 0;
52    SCH_tasks_G[TASK_INDEX].RunMe = 0;
53    return Return_code; // return status
54  }
55
56  void SCH_Dispatch_Tasks(void) {
57    uint32_t Index;
58    // Dispatches (runs) the next task (if one is ready)
59    for (Index = 0; Index < SCH_MAX_TASKS; Index++) {
60      if (SCH_tasks_G[Index].RunMe > 0) {
61        (*SCH_tasks_G[Index].pTask)(); // Run the task
62        SCH_tasks_G[Index].RunMe--; // Reset / reduce RunMe flag
63      }
64    }
65  }
66
67  void SCH_Init(void)
68  {
69    unsigned char i;
70    for(i = 0; i < SCH_MAX_TASKS; i++)
71    {
```

```
    SCH_Delete_Task(i);
  }
}
```

Listing 1: Source code of global.c

Changes in main.c

```
int main(void)
{
  HAL_Init();
  SystemClock_Config();
  SCH_Init();
  MX_GPIO_Init();
  MX_TIM2_Init();
  HAL_TIM_Base_Start_IT(&htim2);

  SCH_Add_Task(fsm_auto, 0, 1);
  SCH_Add_Task(fsm_manual, 0, 1);
  SCH_Add_Task(fsm_setting, 0, 1);

  while (1)
  {
    SCH_Dispatch_Tasks();
  }
  return 0;
```

Listing 2: Apply scheduler for Lab3

2 Link

For more details. please refer to this source code **Link Github**